REMARKS

The enclosed is responsive to the Examiner's Office Action mailed on January 16, 2007. At the time the Examiner mailed the Office Action claims 1-33 were pending. By way of the present response the Applicants have: 1) amended claims 1, 12 and 23; 2) added no new claims. As such, claims 1-33 are now pending. The Applicants respectfully request reconsideration of the present application and the allowance of all claims now represented.

The Examiner rejected claims 1-4, 8, 12-15, 19, 23-26 and 30 under 35 U.S.C. 102(e) as being anticipated by Faraj, U.S. Publication 2002/0073063 (hereinafter "Faraj"); and rejected claims 5-7 and 9-11 under 35 U.S.C. 103(a) as being unpatentable over Faraj, U.S. Publication 2002/0073063 (hereinafter "Faraj"), in view of Berry, et al., U.S. Patent 6,662,359 (hereinafter "Berry").

Applicant respectfully submits that Faraj does not disclose or suggest features recited in the present set of claims. Specifically, with respect to Claim 1, Faraj does not disclose or suggest modifying bytecode associated with the one or more application components, the modifications associated with a particular set of methods of the application components related to program execution across application servers, databases and/or external systems.

With respect to at least a portion of this claim feature, the Office Action cites to the following paragraphs of Faraj:

> [0010] A second approach to automating instrumentation of Java applications is the post-processing of compiled Java byte code. This approach uses byte code manipulation libraries that enable the insertion and modification of byte code in a Java .class file so that the Java code generates an execution trace at runtime. This approach has the following two disadvantages. Firstly, modifying byte code potentially creates Java security problems. When a JAR file (Java archive file) is signed using JDK software tools, each file in the archive is given a digest entry in the archive's manifest. The digest values are hashes or encoded representations of the contents of the files as they were at the time of signing, and they will change if the file itself changes. Verifying a signed JAR file means that the digests of each of its files is to be re-computed and then compared with the digests recorded in the manifest to ensure that the contents of the JAR file haven't changed since it was signed. Hence, byte code modification will result in difficulties where the application to be traced is in a signed JAR file. The application will not pass the security verification step as a result of the modification to the byte files made for tracing purposes.
>
> [0011] In addition, modifying the byte code means that the resulting code can no longer be easily debugged. Debuggers expect a certain match between Java source files and their corresponding compiled class files. When the byte code in a .class file is modified, there is no longer a match between the source code line numbers and the new .class files.
>
> Faraj, paras. [0010]-[0011] (emphasis added).

Notably, this paragraph is from the "Background of the Invention" section of Faraj where Faraj describes the problems and limitations associated with using bytecode modification techniques for performing tracing functions. Thus, Faraj teaches away from the use of bytecode modification techniques as claimed in the present application. Faraj does not describe that these techniques should be used within the context of the invention described in the Detailed Description of Faraj.

Moreover, Faraj does not teach or suggest the use of bytecode modification techniques within the context of a distributed statistical records (DSR) system as claimed in the present application. Applicants have amended the claims to indicate more precisely the types of methods to which the claims are directed. For example, Claim 1 now recites that the bytecode modification techniques are used on "a particular set of methods of the application components related to program execution across application servers, databases and/or external systems" (i.e., methods related to the collection of distributed statistical records).

Berry teaches bytecode modification techniques in which entry and exit "hooks" may be inserted within program code. However, as mentioned above, Faraj teaches away from the use of bytecode modification techniques as claimed in the present application and does not disclose or suggest the use of bytecode modification techniques for methods of application components related to program execution across application servers, databases and/or external systems." Accordingly, Applicant respectfully submits that Claim 1 is in condition for allowance.

Independent Claims 12 and 23 include similar elements as Claim 1. Accordingly, Applicant submits that these claims are in condition for allowance for the same reasons as set forth above. In addition, because claims 2-11; 13-22 and 24-33 depend from independent Claims 1, 12, and 23, respectively, and include additional features, Applicant respectfully submits that these claims are also in condition for allowance.

CONCLUSION

Applicant respectfully submits that all redetections have been overcome and that all pending claims are in condition for allowance.

If there are any additional charges, please charge them to our Deposit Account Number 02-2666. If a telephone conference would facilitate the prosecution of this application, the Examiner is invited to contact Thomas C. Webster at (408) 720-8300.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 5/16/07

Thomas C. Webster
Reg. No.: 46,154

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300